



Augmented models for improving vision control of a mobile robot

Andersen, Gert Lysgaard; Christensen, Anders C.; Ravn, Ole

Published in:
Proceedings of the Third IEEE Conference on Control Applications

Link to article, DOI:
[10.1109/CCA.1994.381270](https://doi.org/10.1109/CCA.1994.381270)

Publication date:
1994

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Andersen, G. L., Christensen, A. C., & Ravn, O. (1994). Augmented models for improving vision control of a mobile robot. In *Proceedings of the Third IEEE Conference on Control Applications* (Vol. Volume 1, pp. 53-58). IEEE. <https://doi.org/10.1109/CCA.1994.381270>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

AUGMENTED MODELS FOR IMPROVING VISION CONTROL OF A MOBILE ROBOT

Gert L. Andersen, Anders C. Christensen, Ole Ravn

Institute of Automatic Control Systems, Bldg. 326
The Technical University of Denmark
DK-2800 Lyngby, Denmark

Abstract

This paper describes the modelling phases for the design of a path tracking vision controller for a three wheeled mobile robot. It is shown, that by including the dynamic characteristics of vision and encoder sensors and implementing the total system in one multivariable control loop, one obtains good performance even when using standard low cost equipment and a comparatively low sampling rate. The plant model is a compound of kinematic, dynamic and sensor submodels, all integrated into a discrete state space representation. An intelligent strategy is applied for the vision sensor, and includes the start up, normal operation, exception handling and shut down phases. Laboratory experiments show the validity of the approach using a standard Kalman Filter / LQR control design.

1. Introduction

One of the topics within robotic research is *computer vision*. The use of vision is motivated by the continuing need to increase the flexibility and scope of applications of robotic systems.

The advances in vision technology, i.e. sensors and image processing equipment, has made it possible to introduce vision sensors into real time control systems. Vision is a multi-purpose sensor as different algorithms can be used during different modes of operation. The functionality of the sensor can be reconfigured on-line to adapt to different modes of operation [1].

It is straightforward, when designing the manoeuvring control for a mobile robot, to use a layered architecture of wheel servo / autopilot / navigation pilot, where wheel velocity servoes are implemented as inner loops, and the autopilot uses a kinematic model only. The problem in this approach is that upper levels demand lower levels to be dynamically negligible: the higher maximal speed of the robot required, the higher demands for the speed and

accuracy of the equipment. This goes for the bandwidth of the wheel servoes, precision of the encoders, image sensing and processing speed. For fairly normal speeds these bandwidth requirements exceed microprocessors of standard price levels.

We design a controller that can keep a mobile robot following a predefined track using vision feedback. In order to minimize cost and complexity of equipment, the design is based on a one-level modelling of the robot and the sensors involved.

The outline of the paper is as follows. First we describe the mobile robot and the special mode used for navigation using vision feedback. Then we derive a discrete-time model, which is a compound model, combining three different submodels: a kinematic, a dynamic and a sensor model. We apply standard methods to carry out the observer and controller design. Finally, we validate the closed loop performance in laboratory experiments.

2. The Mobile Robot

During the last years, a 2 DOF mobile robot has been constructed at the Institute of Automatic Control Systems, see figure 1.

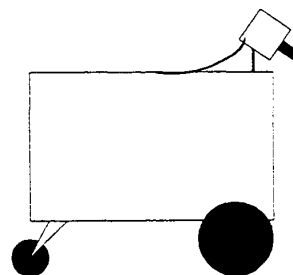


Figure 1. Sketch of the mobile robot at IACS.

The mobile robot serves as a test bed for the comparisons of different algorithms for control of intelligent vehicles on equal terms, without letting the implementation differences play a major role. The robot is built using standard components

instead of buying a ready made vehicle as this gives more control of the degree of flexibility and versatility, [2].

The robot is moved and steered by two independently driven front wheels and at the rear end there is a castor wheel for stability. For the purpose of controlling the basic motions, the robot is equipped with digital encoders on each of the motor shafts, and a CCD camera is placed in front of the robot in a pan and tilt mechanism. The robot's main dimensions are: 70 cm long, 50 cm wide, 55 cm high. The total weight is 70 kg.

The position of the camera, relative to the center point of the vehicle, is defined by the height above floor level, h , and the displacement ahead of the center line, l . The camera orientation is defined by the tilt angle, θ_t , and the pan angle, θ_p , figure 2.

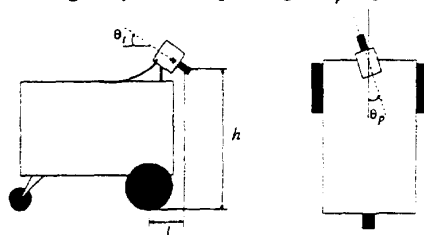


Figure 2. Definition of camera mount.

The Highway Mode

In this paper, the mobile robot is operated in *highway mode*, e.g. it follows a black tape track on the floor using a vision sensor, a vision algorithm, the encoder sensors and a control algorithm.

Stated in automation terms, i.e. start up, normal operation, exception handling, shut down, the *structure* of the vision algorithm can be explained as follows:

During **start up**, the robot is at rest while the overall picture is searched for the track.

During **normal operation**, the robot is moving at maximum velocity. The vision algorithm scans for the track in a line in the bottom of the sensor picture and another in the top, figure 3.

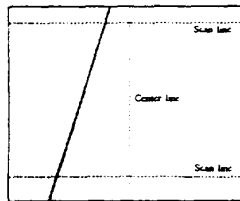


Figure 3. The view of the camera during highway mode. Two horizontal scan-lines and the center line are indicated.

For stopping at intersections and decreasing speed for possible turns, the top line is scanned for intersecting tracks as well.

If the track is changing too rapidly, if the tracking becomes poor, or if the track is lost in the sensor picture, then, the vision algorithm uses its **exception handling** routine. In this particular case, the exception handling routine decreases the velocity of the robot to a predefined level using constant deceleration. When the path tracking is reestablished and the track seems straight, the velocity is increased to maximum again. If the track is lost and not found again, the robot is stopped (**shut down**).

In highway mode, the vision algorithm outputs the lateral position error, found as the distance from the track to the center line of the sensor picture. A picture is scanned at every 40 ms and saved in the computer RAM, then a match filter is used in order to find the track. The lateral position error is found and intersecting tracks are sought for.

3. Modelling

The following models are needed in order to be able to use standard methods for the controller design:

A *kinematic* model which expresses the relationship between the lateral position error and the angular velocity of the robot.

A *dynamic* model which expresses the relationship between linear and angular velocities of the robot, and the input voltages to the DC steering motors.

A *sensor* model which expresses the relationship between the physical quantities being measured and the measurements available from the sensor.

Finally, a *discrete-time* model combining the kinematic, the dynamic and the sensor models.

Kinematic Model

When the kinematic model is derived, i.e. finding the connection between the lateral position error, x_e , and the angular velocity of the robot, ω , the dynamics are neglected. Further it is assumed that the robot moves at a constant speed, V . Following figure 4, some geometric considerations result in the set of equations (1), (2) and (3):

$$x_e = L \tan(\theta) + \frac{x_0}{\cos(\theta)} \quad (1)$$

$$\dot{x}_e = V \sin(\theta) \quad (2)$$

$$\dot{\theta} = \omega = \frac{r}{b} (\omega_r - \omega_l) \quad (3)$$

where ω_r and ω_l are angular velocities of right and left wheel, and r and b are wheel- radius and base, respectively. L is the distance between the point of rotation and the point where the lateral position error is measured, x_0 is the lateral displacement of the robot and θ its orientation relative to the track.

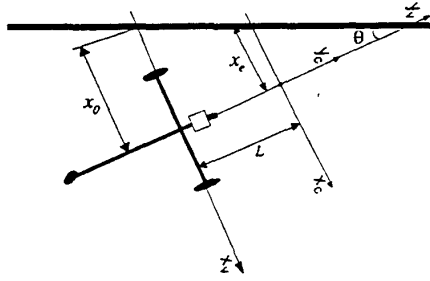


Figure 4. The robot placed close to the track.

Assuming small deviations from $\theta = 0^\circ$, the following matrices in a standard state space description are obtained:

$$A_k = \begin{bmatrix} 0 & 0 \\ V & 0 \end{bmatrix} \quad B_k = \begin{bmatrix} \frac{r}{b} & -\frac{r}{b} \\ 0 & 0 \end{bmatrix} \quad (4)$$

$$C_k = [L \ 1], \quad D_k = [0] \quad (5)$$

where the state and input vectors are:

$$x^T = [\theta \ x_0], \quad u^T = [\omega, \ \omega_l] \quad (6)$$

Dynamic Model

The dynamic model is derived from the theory of rigid systems. A rigid system is defined as a system in which the distance between its parts do not change. Planar motions for rigid systems are described as the sum of translation and rotation. In the following superscripts w and m are referring to wheel and motor, respectively.

The dynamic model is derived in a coordinate system fixed in the vehicle body, see figure 5.

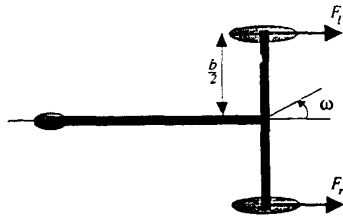


Figure 5. Sketch of a three wheeled robot.

Using Newtons second law around the robots center point the following equations may be derived:

$$M \dot{v} = F_r + F_l \quad (7)$$

$$I \dot{\omega} = \frac{b}{2} F_r - \frac{b}{2} F_l \quad (8)$$

where v and ω denote translational and angular velocities, I the moment of inertia, and M the total

mass of the robot. F_r and F_l are right and left wheel force, respectively.

Solving (7) and (8), the following expressions for the force on the wheels are obtained:

$$F_r = \frac{M}{2} \dot{v} + \frac{I}{b} \dot{\omega} \quad (9)$$

$$F_l = \frac{M}{2} \dot{v} - \frac{I}{b} \dot{\omega} \quad (10)$$

We have the following expressions for the connection between the translational and angular velocities of the center point v and ω and the wheel velocities

$$v = \frac{v_r^w + v_l^w}{2} \quad (11)$$

$$\omega = \frac{v_r^w - v_l^w}{b} \quad (12)$$

Similar expressions are obtainable for the linear and angular acceleration, by simple differentiation.

Likewise, we have the relationships (13) and (14) between angular velocity of the wheel axle ω^w and wheel velocity v^w , and torque τ on the motor shaft and wheel force.

$$v^w = r \omega^w \quad (13)$$

$$\tau = \frac{r}{N} F \quad (14)$$

where r is the wheel radius and N , is the gear ratio between the motor and wheel axle.

Inserting the expressions for v , ω , v^w , and τ in the wheel force equations (9) and (10), one obtains

$$\begin{aligned} \tau_r^m &= \frac{r^2}{N} \left(\frac{M}{4} + \frac{I}{b^2} \right) \dot{\omega}_r^w + \frac{r^2}{N} \left(\frac{M}{4} - \frac{I}{b^2} \right) \dot{\omega}_l^w \\ &= k_r \dot{\omega}_r^w + k_{lr} \dot{\omega}_l^w \end{aligned} \quad (15)$$

$$\begin{aligned} \tau_l^m &= \frac{r^2}{N} \left(\frac{M}{4} - \frac{I}{b^2} \right) \dot{\omega}_r^w + \frac{r^2}{N} \left(\frac{M}{4} + \frac{I}{b^2} \right) \dot{\omega}_l^w \\ &= k_{rl} \dot{\omega}_r^w + k_l \dot{\omega}_l^w \end{aligned} \quad (16)$$

expressing the relation between the wheel angular accelerations and the torque on the motor shafts.

It is now possible to obtain a model, expressing the dynamic behaviour from the input to the DC - motors to the wheel velocities, that includes the coupling between the two driven wheels.

Figure 6 shows a block diagram for the two DC - motors including the cross couplings k_{lr} and k_{rl} (k_r , k_l , R and J are the motor constants).

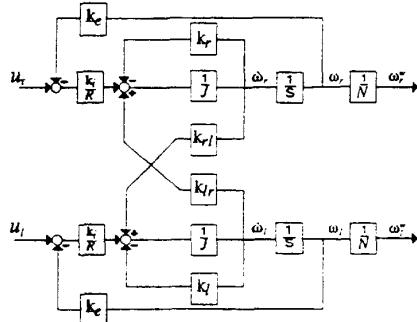


Figure 6. Block diagram representing the robot dynamics. Subscripts r and l are omitted for the motor constants.

Choosing as state vector the angular velocities of the wheel axes and as input vector the ancre voltage to the DC - motors

$$x^T = [\omega_r, \omega_l], \quad u^T = [u_r, u_l] \quad (17)$$

the matrices in a standard linear state space model is given by equations (18) - (20) below.

$$A_d = \begin{bmatrix} \frac{-(J_l + \frac{k_l}{N})k_r k_d}{R_l(J_l + \frac{k_l}{N})(J_l + \frac{k_l}{N}) - \frac{k_l k_d}{N^2}} & \frac{-\frac{k_r k_d}{N}}{R_l(J_l + \frac{k_l}{N})(J_l + \frac{k_l}{N}) - \frac{k_l k_d}{N^2}} \\ \frac{-\frac{k_r k_d}{N}}{R_r(J_l + \frac{k_l}{N})(J_l + \frac{k_l}{N}) - \frac{k_l k_d}{N^2}} & \frac{-(J_r + \frac{k_r}{N})k_l k_d}{R_r(J_r + \frac{k_r}{N})(J_r + \frac{k_r}{N}) - \frac{k_r k_d}{N^2}} \end{bmatrix} \quad (18)$$

$$B_d = \begin{bmatrix} \frac{(J_l + \frac{k_l}{N})k_r}{NR_l(J_l + \frac{k_l}{N})(J_l + \frac{k_l}{N}) - \frac{k_l k_d}{N^2}} & \frac{\frac{k_r k_d}{N}}{NR_l(J_l + \frac{k_l}{N})(J_l + \frac{k_l}{N}) - \frac{k_l k_d}{N^2}} \\ \frac{\frac{k_r k_d}{N}}{NR_r(J_r + \frac{k_r}{N})(J_r + \frac{k_r}{N}) - \frac{k_r k_d}{N^2}} & \frac{(J_r + \frac{k_r}{N})k_l}{NR_r(J_r + \frac{k_r}{N})(J_r + \frac{k_r}{N}) - \frac{k_r k_d}{N^2}} \end{bmatrix} \quad (19)$$

$$C_d = [r/2 \quad r/2], \quad D_d = [0 \quad 0] \quad (20)$$

Combining the kinematic and dynamic models (4) - (6) and (17) - (20), one obtains the following augmented continuous-time model,

$$\begin{aligned} u^T &= [u_r, u_l] \\ x^T &= [\omega_r, \omega_l, \theta, x_0] \\ y^T &= [V, x_s] \end{aligned} \quad (21)$$

$$A = \begin{bmatrix} A_d & 0 \\ B_s & A_s \end{bmatrix}, \quad B = \begin{bmatrix} B_d \\ 0 \end{bmatrix} \quad (22)$$

$$C = \begin{bmatrix} C_s \\ C_c \end{bmatrix} = \begin{bmatrix} C_d & 0 \\ 0 & C_k \end{bmatrix}, \quad D = \begin{bmatrix} D_s \\ D_c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (23)$$

Sensor Model

In order to measure velocity and lateral position error, the robot is equipped with two different types of digital sensors. The first type is a standard surveillance camera with a frame grabber and an onboard computer. The second type is a set of digital encoders placed on each of the motor shafts.

The Vision Sensor. To perform mobile robot navigation from a robot mounted camera, it is important to be able to interpret the locations of features observed in the image. This involves determining both the intrinsic camera parameters and the positioning of the camera mount with respect to the robot.

A simple linear model for the relationship between distances measured in world coordinates, (x, y) , and the result in camera coordinates, $(\Delta x, \Delta y)$, is:

$$\begin{aligned} \frac{x}{L} &= \frac{\Delta x}{f_x} \\ \frac{y}{L} &= \frac{\Delta y}{f_y} \end{aligned} \quad (24)$$

This model is described by 4 intrinsic parameters: the focal lengths in vertical and horizontal direction, (f_x, f_y) , and the position of the center point in the picture. L is the distance between the object and picture plane, i.e. the distance between the camera and the object. The intrinsic parameters are fixed for a certain adjustment of the camera (zoom and focus) and are found by a standard calibration method. The variation of the intrinsic parameters, due to distortion, is less than 5% [3] and can thus be neglected.

Due to the sensing, preprocessing and segmentation processes, the camera sensor introduces a time delay into the control loop. In the present implementation, the delay is constant and equals $\Gamma_d = 35$ ms, as opposed to a sampling period of $T = 80$ ms.

The presence of the camera results in an augmentation of the plant model. Assuming a continuous, time invariant system $[A, B, C]$ with a zero-order hold at the input, and a time delay of a fraction $0 < d < 1$ of the sampling period T , one obtains

$$\begin{aligned} x(k+1) &= A^* x(k) + B^* u(k) \\ y(k) &= C_{c1}^* x(k) + C_{c2}^* u(k-1) \end{aligned} \quad (25)$$

where A^* , and B^* denotes the discrete representations of the continuous system and C_{c1} and C_{c2} are

$$C_{c1}^* = \frac{f_x}{L} C e^{-\lambda d T}, \quad C_{c2}^* = -\frac{f_x}{L} C e^{-\lambda d T} \int_0^{dT} e^{\lambda \tau} B d\tau \quad (26)$$

with f_x and L defined as in (24).

A standard state space form is obtained by representing the delayed control signal as a state:

$$\begin{bmatrix} x(k+1) \\ x_r(k+1) \end{bmatrix} = \begin{bmatrix} A^* & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x_r(k) \end{bmatrix} + \begin{bmatrix} B^* \\ I \end{bmatrix} u(k) \quad (27)$$

$$y(k) = \begin{bmatrix} C_{cl}^* & C_{cz}^* \end{bmatrix} \begin{bmatrix} x(k) \\ x_r(k) \end{bmatrix}$$

The Digital Encoder. The mobile robot has digital encoders placed directly on each of the motor shafts. The encoders work as reset integrators, i.e they are read at each sampling instance and then reset to zero. This measurement expresses the distance driven during one sampling period, and can be interpreted as an average velocity.

A reset integrator is represented in the following way:

$$z(k) = \int_0^T y(\tau) d\tau \quad (28)$$

Here $y(\tau)$, $k < \tau < k+1$ can be expressed as

$$y(\tau) = C e^{A(\tau-k)} x(k) + C \int_0^{\tau-k} e^{A(\tau-s)} B ds u(k) \quad (29)$$

assuming that the system $[A, B, C]$ is continuous, linear, time invariant, and includes a zero-order hold at the input.

Combining (28) and (29), and assuming a sampling period T results in

$$z(k) = A_z^* x(k) + B_z^* u(k) \quad (30)$$

where A_z^* and B_z^* are given by:

$$A_z^* = \int_0^T C e^{A\tau} d\tau \quad (31)$$

$$B_z^* = \int_0^T \int_0^{\tau} C e^{A(\tau-s)} B ds d\tau \quad (32)$$

A digital encoder model introduces a new state into the resulting discrete-time system. The augmented system can be expressed, assuming $D^* = 0$, as

$$\begin{bmatrix} x(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} A^* & 0 \\ A_z^* & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} B^* \\ B_z^* \end{bmatrix} u(k) \quad (33)$$

$$y(k) = \begin{bmatrix} 0 & C_z^* \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix}$$

where $[A^*, B^*, D^*]$ is the discrete representation of the continuous-time system.

Referring to the continuous-time plant equations above, we measure the translational velocity of the robot by applying the encoders to right and left shaft, respectively (k_{enc} is the encoder gain):

$$\begin{bmatrix} z_r(k) \\ z_l(k) \end{bmatrix} = N k_{enc} \int_0^T \begin{bmatrix} \omega_r(\tau) \\ \omega_l(\tau) \end{bmatrix} d\tau \quad (34)$$

thus obtaining

$$V(k) = \begin{bmatrix} r & r \\ 2 & 2 \end{bmatrix} \begin{bmatrix} z_r(k) \\ z_l(k) \end{bmatrix} \quad (35)$$

hence,

$$C_z^* = \begin{bmatrix} 0 & r & r \\ 2 & 2 & 2 \end{bmatrix} \quad (36)$$

Resulting Discrete Model

Combining the different models described above and the sampling process results in a linear time invariant discrete state space model of 8th order.

$$\begin{aligned} u_a^T &= [u_r \quad u_l] \\ x_a^T &= [x \quad x_r \quad z] \end{aligned} \quad (37)$$

$$A_a^* = \begin{bmatrix} A^* & 0 & 0 \\ 0 & 0 & 0 \\ A_z^* & 0 & 0 \end{bmatrix} \quad B_a^* = \begin{bmatrix} B^* \\ I \\ B_z^* \end{bmatrix} \quad (38)$$

$$C_a^* = \begin{bmatrix} C_{cl}^* & C_{cz}^* & 0 \\ 0 & 0 & C_z^* \end{bmatrix} \quad D_a^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (39)$$

The matrix exponential integrals in (26), (31) and (32), are solved using a result from [4].

To ensure that the velocity of the robot has zero steady state error, the system is augmented with an integral state, resulting in a 9th order model.

The system is completely observable and controllable. The A matrix is singular due to the extra states introduced by the camera and encoder models.

4. Design of vision controller

This section summarizes briefly the controller design. The controller is based on state feedback and is designed using standard LQ techniques.

Kalman Filter

For state reconstruction an optimal Kalman filter has been designed. The Kalman filter is implemented in a way that uses the latest measurements for state update, [5].

The state and measurement noise is due to the quantization in the DA converter, the camera and encoders, respectively. Quantization noise can be approximated by uncorrelated discrete-time white noise with variance $h^2/12$, h being the smallest representable unit, [6]. The resolution of the DA converter, camera, and encoders equals $h_{DA}=5.0\text{mV}$, $h_{cam}=1.0\text{mm}$, $h_{enc}=0.7\text{mm}$, respectively.

Controller

The controller gains are found using LQR techniques. The states and inputs (21) are weighted using diagonal weight matrices in continuous time. These are then transformed to equivalent discrete weight matrices. The discrete-time weight matrices include cross product terms that penalize the product of the states and inputs, [7].

5. Performance

To validate the performance of the closed loop control system and the applied sensor strategy the robot is commanded to follow a tape track, having two lateral shifts, figure 7. The slope of the shifts equals -0.6 and 0.35 respectively. During the experiment the vision algorithm varies the velocity of the vehicle between $0.3 - 0.8 \text{ m/s}$ with the acceleration fixed to 0.5 m/s^2 .

Figure 7 shows a reconstruction of the robot motion based on physical encoder measurements, using a simple robot simulator. Figure 8 shows the path tracking, i.e. the lateral position error measured by the camera. Figure 9 and 10 shows the translational and rotational velocity of the robot.

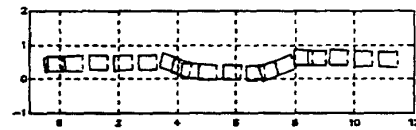


Figure 7. Robot movements in a (x [m],y [m]) frame.



Figure 8. Lateral position error. Units: [m] versus [m].

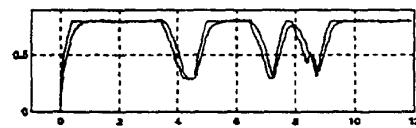


Figure 9. Translational velocity. Units: [m] versus [m/sec].



Figure 10. Angular velocity. Units: [m] versus [rad/sec].

The system keeps the tracking error within $\pm 0.02\text{m}$, except when the reference is changing. As seen from figures 7 and 10 the robot moves smoothly, showing no tendency to oscillate, keeping in mind that the robot drives between 6.4 cm and 2.4 cm during each sample.

6. Conclusion

The work described in this paper shows how well known standard modelling and design techniques can be combined to solve the path tracking problem using vision feedback. We have obtained a closed loop control system that performs well, despite of low cost equipment, (standard surveillance camera, digital encoders, low sampling rate).

The keys to succes has been a proper modeling of the dynamical system including the applied sensors and the application of an intelligent strategy for the vision sensor.

Acknowledgements

The basic vision algorithms are implemented by Allan Holmegaard and described in [8].

References

- [1] Andersen, N.A., Ravn, O., Sørensen, A.T., "Real-time vision based control of servomechanical systems", *Proc. of the Second International Symposium on Experimental Robotics*, France, 1991.
- [2] Ravn, O., Andersen, N.A., "A test bed for experiments with intelligent vehicles", *Proc. of 1st IFAC international Workshop on Intelligent Autonomous Vehicles*, Hampshire, United Kingdom, 1992.
- [3] Ravn, O., Andersen, N.A., Sørensen, A.T., "Auto-calibration in automation systems using vision", *Proc. of the Third International Symposium on Experimental Robotics*, Japan, 1993.
- [4] Loan, C.F.V., "Computing integrals involving the matrix exponential", *IEEE Transactions on Automatic Control*, vol AC-23, no. 3, 1978.
- [5] Kwakernaak, H., Sivan, R., *Linear optimal control systems*, Wiley-Interscience, 1972.
- [6] Levin, M.L., *Datamatbaserede reguleringssystemer*, Servolaboratoriet, DTH, (in danish), 1982.
- [7] Franklin, G.F., Powell, J.D., Workman, M.L., *Digital control of dynamic systems*, Second Edition, Addison-Wesley Publishing Compagny, 1990.
- [8] Holmegaard, A.Z., *Robust regulering af AGV*, Master Project, Servolaboratoriet, DTH, 1992.